How to Share and Own a Secret

Victor Ermolaev[1] and Gamze Tillem[1]

ING Bank, The Netherlands
{victor.ermolaev,gamze.tillem}@ing.com

Abstract. Custodian service is a service safeguarding a firm's or individual's financial assets or secret information. Such services often present a user with a security versus ownership dilemma. The user does not wish to pass the full control over their asset to the custodian to facilitate safeguarding. A control sharing mechanism allowing the custodian to hold enough information and keeping the user as the owner of the asset is required. For the assets being secret information, cryptographic protocols addressing this dilemma are known as prepositioned secret sharing (PSS) protocols. PSS schemes distinguish redundant "common" shares and a specific "activating" shares controlling the very possibility of the secret information reconstruction. Usually PSS schemes: 1) lack robustness with respect to the amount of "common" shares, i.e., a high redundancy degree in "common" enables them to reconstruct the secret without "activation", and 2) are inflexible in configuring the robustness of the "activating" shares, i.e., how many "activating" shares can be lost or stolen before the secret can be reconstructed. In this paper, we present a PSS addressing these shortcomings.

Keywords: Secret-sharing · Cryptology · Security Protocols

1 Introduction

In his seminal work, Shamir [21] introduced a secret-sharing scheme based on an evaluation of polynomial of degree k in n points and consequent reconstruction of the secret free coefficient by any subset $|T|$ of the n points, $k-1 \leq t \leq n$. Such schemes are commonly referred to as (t, n) sharing schemes.

Consequent work of Simmons [22, p. 393-394] identified conceptual capabilities that these sharing schemes must offer to be used in "real" (sic) applications. Our work addresses two challenges from that list. For the sake of completeness these two items are quoted entirely.

1. Prepositioned shared secret schemes in which the holders of the private pieces of information are unable to recover the secret information, even if they all collude to do so, until such time as the scheme is activated by communicating additional information.
2. Prepositioned shared secret schemes in which the same collection of private pieces of information can be used to reveal different secrets depending on the choice of the activating information.

There are several areas of applications for PSS schemes. Simmons [23] provides many examples of military applications. Another particularly need for such schemes exists in custodian services. A custodian is an institution that holds customers' secrets for safekeeping to prevent these secrets from being stolen or lost. A prominent example is a bank holding customers' securities. In traditional custodian model, a customer transfers full authority over their assets to the custodian. To make this model viable, either a complete trust relation between the custody service providers and their users must be assumed, or, when the trust is lacking, a legal framework holding the custodian accountable must fill the void. Prepositioned shared secret schemes have a potential to remove the "must" restriction in the latter case. A constellation of custodian institution holds shards of secret which can only make up a secret if additional customer information is provided, but by itself the customer's information is as useless as the aforementioned shards.

Our work proposes a PSS based on polynomial evaluation and reconstruction akin to the Shamir secret sharing. It allows for two classes of participants (the custodians and the user) that need to engage into an information exchange to reconstruct the secret. At the same time neither of the classes holds sufficient information to reconstruct the secret on their own. Our PSS allows the custodians to increase own robustness by generating redundant information pieces on their own. Such redundant pieces do not compromise the above guarantee. Similarly, the user may also increase their robustness to a certain limit (which is covered further on in the paper).

2 Prepositioned Secret Sharing

2.1 Activating Information

The original challenges (Enumeration 1) neither define "activating information", nor speak on its nature. This freedom allows different interpretations and leaves a lot of room for a debate. It is tempting to define "activating information" in terms of external factors, especially in the light that the choice of such information can reveal different secrets. In consequent research such information is often chosen to be of internal nature: Xu et al. [25] utilize n-of-n Shamir sharing and define "activating information" as one special share, in Eskicioglu [10] "activating information" is defined as a subset of Shamir shares. We also define such information as some shares of an appropriate Shamir sharing. The sharing scheme is presented below.

2.2 Sharing Scheme

Let parties $U_i, i = 1, \ldots, t$ be providers of activating information and parties $C_i, i = 1, \ldots l$ be custodians. We shall also refer to the holders of activating information as users or activators. Sharing and reconstruction is facilitated by a polynomial of degree m, $f(x) = \sum_{k=1}^{m} a_k x^k + b$. Let f be called an *instrumental*

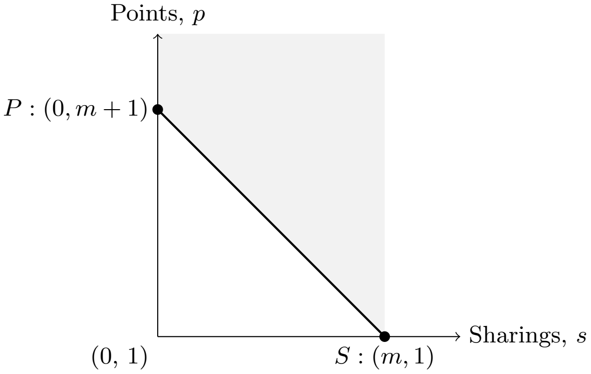

Fig. 1. Segment PS is defined as $p=(m+1)-s$. The grey area contains valid setups for the scheme.

polynomial and set of its coefficients be denoted with $A=\{a_k\}_{k=1}^m$. Polynomial f can be defined both algebraically, as above, and geometrically, by specifying points $(x_i, f(x_i)), i \geq m+1$. Note that all the polynomial operations performed in the following sections are performed on a finite field. The notation is disregarded for brevity.

Sharing Now we are ready to define the sharing scheme:

1. fix the secret b, construct any $f(x)=\sum_{k=1}^m a_k x^k + b$, now the secret is $f(0)$,
2. select a subset $\widehat{A} \subseteq A$, let $|\widehat{A}| =: s$, where $|\cdot|$ denotes set cardinality; consider each $a_k \in \widehat{A}$ to be secret by itself and share it under (t_k, n_k) Shamir secret sharing, let us call this sharing instance $S_k = (a_{k,j} \mid j = 1 \ldots n_k)$, where $a_{k,j}$ is the j-th share of the k-th coefficient. Distribute the shares in *some* way to the custodian parties,
3. let $p=(m+1)-s$, generate a set of distinct points

$$\begin{aligned} &V=\{(x_j, f(x_j)) \mid x_j \neq 0,\ j \geq p\}, \\ &x_j \neq x_i, \forall x_j, x_i \in V \end{aligned} \tag{1}$$

   Distribute the shares in *some* way to the future holders of the activating information.

The relation between the number of the sharing instances and the number of points is illustrated in Fig. 1. The abscissa shows how many coefficients have been distributed to the custodians and the ordinate – how many points must be held by the activators. If one chooses to have 0 sharings, then the activating information reduces to the classical Shamir secret sharing scheme (and will require $m+1$ points). Oppositely, if the number of sharings is equal to the degree of the instrumental polynomial m, the scheme is a prepositioned sharing scheme

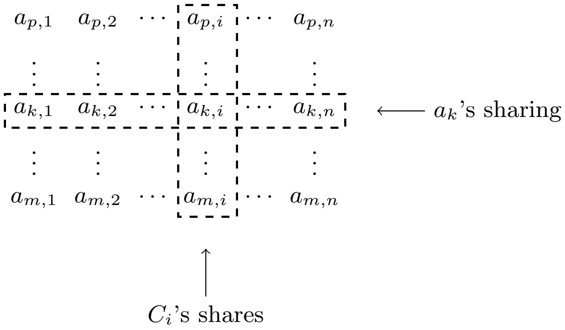
$a_{p,1}$ $a_{p,2}$ $\cdots$ $a_{p,i}$ $\cdots$ $a_{p,n}$
$a_{k,1}$ $a_{k,2}$ $\cdots$ $a_{k,i}$ $\cdots$ $a_{k,n}$
$a_{m,1}$ $a_{m,2}$ $\cdots$ $a_{m,i}$ $\cdots$ $a_{m,n}$
a_k's sharing
C_i's shares

Fig. 2. Custodian sharing matrix.

in the sense of [22] with a single piece activating information. With these edge cases, we deduce that the origin of the grid is $(0, 1)$.

The section PS depicts the minimal amount of points required given the number of the shared coefficients, e.g., given a quadratic instrumental polynomial and fixing the number of the shared coefficients to 1 will require at least 2 points to reconstruct the secret. Relation $p = (m + 1) - s$ bounds from below the number of necessary points; redundant points can be generated to increase the activators' robustness (we shall discuss this in Section 2.3), thus any integer-valued coordinate from the grey area is a valid setup for the scheme.

Steps (2) and (3) of the scheme offer a great amount of flexibility allowing to configure the scheme to a desired number of parties and levels of robustness. To simplify the exposition, without loss of generality, some parameters and the shares distribution principle can be adjusted as follows:

2a Let us share the first s greatest coefficients, then $\widehat{A} = \{a_{m-s+1}, \ldots, a_m\}, 0 \leq s \leq m$, for $s = 0$, let the set be empty. Equivalently, $\widehat{A} = \{a_p, \ldots, a_m\}$, for $p = 1, \ldots, m + 1$, let the set be empty for $p = m + 1$.
Let $t_k = t, n_k = n$ for $k = p \ldots, m$ and fix the number of custodians to n. Distribute a collection of shares $a_{k,i}, i = 1, \ldots n$ to party C_i. Now each custodian has a shard from all coefficient sharings.
This composition of the coefficients' sharings and distribution can be conveniently represented as a (custodian sharing) matrix, Fig. 2.

3a Activators $U_i, i = 1, \ldots, t$ can be viewed as one generalized activator/user U holding the full set V.

In what follows, the sharing will be meant in the sense of (2a) and (3a).

Secret Reconstruction

1. Recover coefficients a_k, $k = p, \ldots, m$ from the respective sharing instances S_k, denote the recovered versions as $\widetilde{a_k}$.

2. Activate the scheme by injecting additional information from V by providing points $(x_j, f(x_j))$, $j = 1, \ldots, p$ points, solve a system of linear equations for unknown coefficients a_k, $k = 1, \ldots, p$ and b:

$$\begin{cases} b + \sum_{k=1}^{p-1} a_k x_1^k + \sum_{k=p}^{m} \widetilde{a_k} x_1^k = f(x_1) \\ \qquad\qquad \cdots \\ b + \sum_{k=1}^{p-1} a_k x_p^k + \sum_{k=p}^{m} \widetilde{a_k} x_p^k = f(x_p) \end{cases} \tag{2}$$

   Solutions for the coefficients have an auxiliary role for reconstruction of b and thus can be discarded directly.

Theorem 1. *Linear system (2) has a unique solution.*

Proof. Define

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & \cdots & x_1^{p-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_p & \cdots & x_p^{p-1} \end{bmatrix}, \ a = \begin{bmatrix} b \\ a_1 \\ \vdots \\ a_p \end{bmatrix}, \ c = \begin{bmatrix} f(x_1) - \sum_{k=p}^{m} \widetilde{a_k} x_1^k \\ \vdots \\ f(x_p) - \sum_{k=p}^{m} \widetilde{a_k} x_p^k \end{bmatrix}, \tag{3}$$

then linear system (2) can conveniently be written in a matrix form

$$\mathbf{X}\, a = c \tag{4}$$

We need to prove that the $\mathbf{X}$ is invertible. The inverse exists if the matrix determinant is not equal to zero. Observe that $\mathbf{X}$ is a square Vandermonde matrix with

$$\det(\mathbf{X}) = \prod_{1 \le j < i \le p} (x_i - x_j), \tag{5}$$

which is never zero by construction of the set V.

Observe that activation may occur both on the custodian and the user sides. Both approaches have their benefits and concerns, which we shall discuss later in Section 4.

2.3 Scheme Properties

Our concise scheme addresses the challenges outlined for prepositioned shared secret schemes.

- by construction, no subset of shares from $\bigcup_{i:a_i \in \widehat{A}} S_i$ is sufficient to reconstruct the secret b,
- activation is triggered by choosing and providing the necessary amount of points from V,

- same instances $S_i, i = 1, \ldots, p$ of the Shamir secret sharings can be used to share different secret b's. In such scenario, to re-use the custodian information for a new secret, the user needs to
  1. reconstruct the secret as described in Enumeration 2.2, which has a side-effect of reconstructing of the original instrumental polynomial $f(x)$,
  2. replace the secret b in $f(x)$,
  3. generate an appropriate V corresponding to the new secret.

Additionally, the scheme design provides robustness guarantees w.r.t. the amount of shred information and the participants behavior.

Infinite custodian robustness. By construction, the scheme allows the custodian constellation to increase own robustness infinitely by increasing the number of held shares without creating a risk of an unsolicited secret reconstruction. This correspond to adding more columns to the custodian sharing matrix, Fig. 2. The information held by the constellation will *always* require at least an additional piece of information held the user.

Controllable user robustness. User's resistance against loss and theft of shares directly depends on the degree of the instrumental polynomial m. Provided that the number of the shared coefficients $s \neq 0$, number of the user-held shares may reach m. With this respect, it is sensible to choose $m \geq 2$ and for the user to hold any number of shares p, $2 \leq p \leq m$. In such a case, p specifies the extent to which the scheme can withstand loss or theft from the user's side. At worst, all shares $p = m$ are compromised, the attacker cannot reconstruct the secret without engaging with the custodian.

Non-Collusion. Maintaining the right amount of redundant information is a balancing act between collusion facilitation and robustness improvements. One way to achieve both goals is to require a guaranteed honest majority of parties holding the redundant shares [2,5,20]. This requirement is difficult to enforce in the technological domain. In our proposal, large amounts of redundant information only increase the overall robustness and have no influence on the collusion risks. Under no circumstances can either a single custodian or the constellation of custodians obtain the secret without interacting with the activators.

3 Example

Consider a user who wants to share a secret value $b_1 = 2$ without losing the ownership of it.

For the sake of exposition we consider the instrumental polynomial over $\mathbb{Z}_{11}$

$$f(x) = 4x^3 + 3x^2 + 2x + b_i \mod 11. \tag{6}$$

Let $\widehat{A} = \{a_3 = 4, a_2 = 3\}$ (consequently, $s = 2$ and $p = 2$), then

$$\begin{aligned} V_i =& \{(1, 9 + b_i \mod 11), (-1, -3 + b_i \mod 11)\}, \\ V_1 =& \{(1, \qquad 0), (-1, \qquad 10)\}. \end{aligned} \tag{7}$$

Let there be 4 custodians, thus every coefficient in $\widehat{A}$ must be shared under the same $(\cdot, 4)$-Shamir secret sharing, where the minimal amount of parties is a free parameter.

Next, the user chooses to reconstruct the secret, then, after the custodians have recovered a_3 and a_2, using the set V_1 the following system of linear equations is solvable:

$$\begin{cases} f(1) := \quad 4 + 3 + a_1 + b_1 = 0 & \mod 11 \\ f(-1) := -4 + 3 - a_1 + b_1 = 10 & \mod 11 \end{cases} = \begin{cases} a_1 + b_1 = 4 & \mod 11 \\ a_1 - b_1 = 0 & \mod 11 \end{cases}$$

solving to $a_1 = 2$ and $b_1 = 2$.

Next, the user wants confidence that the loss of either share of V_1 will not result in a complete secret loss. They create for $b_1 = 2$ a redundant share $(2, f(2)) = (2, 6)$ and store this share in a cold storage. Now neither the loss of any of shares in V_1 is critical, nor the unlawful extraction of the share from the cold storage allows the attacker to successfully collude with the custodians to recover the secret.

Finally, the user want to store another secret $b_2 = 10$; having the knowledge on all coefficients, the activation set is easily constructable and is

$$V_2 = \{(1,\ 8),\ (-1,\ 7)\} \tag{8}$$

4 Variants of the Scheme

As has been noted before, the scheme is agnostic w.r.t. the activation side. In the following, we discuss different activation variants of the prepositioned secret sharing.

4.1 PSS with Activation on the User Side

If the purpose of the PSS scheme is to reveal the secret value, then as the owner of secret value, the user should be in charge of activation. In such a scenario, the custodians combine their secret shares which reveal the coefficients of the instrumental polynomial and share the result with the user. The user can then reveal the secret value by inputting their private shares on the instrumental polynomial, whose coefficients' subset is already constructed.

In this setting, the proposed scheme serves the purpose of removing the single point of failure by splitting the secret into secret shards. After using the secret, the user can destroy it and request its reconstruction from the custodians the next time it is needed[1]. It is also possible for the user to keep the secret in its local storage and request the reconstruction only if they lose the access to the secret.

[1] Interestingly, in Shamir's seminal work [21], the secret has been constructed and destroyed after use in a lock.

This approach allows to perform the reconstruction in a plain-text domain as no extra privacy requirements are in place. There may exist additional benefits depending on the nature of the secret. For instance, if the secret is a private key, the user may directly use it for signing and decryption. However, there are two prominent downsides to this approach: 1) the user must be sufficiently knowledgeable and technologically-enabled to carry out the aforementioned mathematical/cryptographic operations, 2) the attack focus shifts on the user, which is a supposedly less protected party, and thus the chance of a successful attack increases.

The latter downside may be aided by employing a homomorphic cryptosystem (or a combination thereof). Unfortunately, absence of efficient fully homomorphic cryptosystems significantly reduces the range of possible computations involving the secret. Although the existing cryptosystems [18,11,6] may support a selection of the computations, they increase the scale of the former concern.

4.2 PSS with Activation on the Custodian Side

Opposite to the activation on the user side scenario, the user's share is sent to the custodians so that they can reconstruct the secret and use it on the user's behalf for some operation. This model puts very little technological demands on the user and provides them with "X-as-a-service", e.g., if the secret is a private key, then the "X" service can be data signing.

Unfortunately, in such a scenario, the user loses the control of their private information: a malicious custodian can use this information later to misuse the secret without the user's consent. Clearly, this approach stands a need of an appropriate protocol which would conceal the activating information preventing its re-use by dishonest custodians. Such protocols are commonly referred to as "multi-party computations" and allow mistrusting parties to execute a joint computation without revealing own inputs [7,13,16,17]. Such protocols, to guarantee own security, use cryptographic techniques to conceal a computation and to only reveal the public result of a computation, instead of re-constructing the secret and using it in the computation.

An example use case for this type of computation is threshold signatures, where a digital signature on a message is computed without revealing the private key to the participants. The use of threshold signatures has gained significant attention in blockchain community in the last couple of years and several protocols that transforms existing digital signature algorithms, e.g. ECDSA, EdDSA, to threshold variants are proposed [1,4,8,13,14,16]. The proposed PSS scheme is also suitable to be used as a primitive in threshold signature protocols.

In threshold signatures, the key generation and the signing steps are performed in a multiparty setting whereas the verification step can be performed by anybody who has access to the public key. Consider, for instance, the threshold ECDSA protocol proposed in [13]. The key generation operation of ECDSA generates a private key $x \in_R Z_q$ and a public key $y = g^x \in \mathcal{G}$. A signature σ on a message M can be computed as

$$\begin{aligned} m &= H(M) \text{ and } k \in_R Z_q \\ R &= g^{k^{-1}} \in \mathcal{G} \text{ and } r = H'(R) \in Z_q \\ s &= k(m + xr) \mod q \\ \sigma &= (r,\, s). \end{aligned} \tag{9}$$

To sign the message, each participant should perform computations on their shares such that

$$(r,\, s) = (H'(g^{k_a^{-1}}),\, k_a(m + b_a r)), \tag{10}$$

where subscript "a" refers to the individual share of each participant for the corresponding value. The challenge in the computation is performing multiplication and exponentiation operations by k_a, which is solved with a share conversion protocol in [13]. Using our scheme, the signature can be reconstructed in a similar way given that both the secret x and k are shared and reconstructed using our proposed secret sharing scheme.

The aforementioned threshold signature scheme requires the user's involvement in the computations to prevent leakage of the secret to custodians. To move the full workload on the custodian side, a concealing cryptographic mechanism, such as homomorphic encryption, for the user's shares can be considered. Designing such a protocol goes beyond the scope of this paper and is a subject for future work.

4.3 What is a Successful Attack?

Definition of a successful attack very much depends on the form of the activation. Unfortunately, without full knowledge of actual protocols enabling the activation, we cannot fully describe the attack surface.

Nevertheless, considering the scheme statically, i.e., no activation is happening, and assuming the attacker is different from the custodian, a successful attack *must* amount to two successful attacks on both the custodian and the user. While the exact strategy to minimize this risk depends significantly on the application domain and the technological capabilities of the involved participants, general guidelines can be drawn:

1. Limit the information on the user's side to an acceptable minimum.
2. Involve several custodians with *disjoint* infrastructures[2]. Different custodians can either maintain jointly a secret coefficient or separately different coefficients. Combinations thereof are also possible.

[2] Requirement of having disjoint infrastructures influences only the robustness of the scheme against share losses.

5 Related Work: Comparison of Safe-Guarding Approaches

Ultimately, there are three modes for control distribution over the secret: custodian is in full control, user in full control, and shared control. The custodian is in full control if they enjoy the full trust from the user side and are entirely liable for the user's assets. We disregard this option as it is the very problem we are addressing. The user is in full control if they are solely responsible for safeguarding the secret. The usual way to competently exercise full control over the secret is to personally hold it together with backups. Our scheme also allows the user to stay in full control as an edge case: set $j = m + 1$ for V, now the user has sufficient shares to reconstruct the secret by themselves. We shall not consider these approaches because they clearly identify the user as a primary attack target. Instead we focus on techniques enabling shared control over the secret, i.e. the custodian and the user hold shares of the secret such that neither of them has sufficient information to reconstruct the secret, it can only be done in a collaborative fashion. We review relevant methods and compare them to our proposition.

PSS based on Shamir Secret Sharing (SSS). One approach to realize the safeguarding scenario is to distribute regular Shamir shares in amounts insufficient for the reconstruction to participants, e.g., for a polynomial of degree one, generate two points and distribute one share to the custodian and one to the user. Now the user can activate the secret reconstruction by communicating own share [25]. Not only does this method not constitute prepositioned sharing (only a single secret can be recovered), but is also extremely brittle, loss of a single share compromises the whole arrangement. Another (simplified) approach is to give the custodian one point from a line, then let the user generate another point; the user, by communicating different points, will reveal different secrets [10,9]. Clearly, additional efforts are required to make this concept robust. There are two possible avenues allowing to increase robustness: 1) give more shares to the user, 2) give more shares to the custodian. While the former certainly increases the scheme robustness, it undesirably puts the user in full control. The latter option allows the custodian to accumulate enough shares to gain the full control over the secret reconstruction, which would be inadmissible. These problems stem from the fact that SSS treats all users equally which might not be ideal for a scenario that needs approval of a certain party. Any subset of threshold amount of parties can reveal the secret without seeking for the approval of the corresponding party. Our scheme clearly distinguishes two classes of participants and enables robustness configuration within each class without creating a possibility of unsolicited reconstruction of the user secret.

Geometric PSS's Simmons [23,22] describes general construction of PSS schemes based on defining two algebraic varieties: the domain variety V_d – a collection of points any of which can be the secret, and the indicator variety V_i – a set of points "pointing" (sic) to the secret in V_d. "Pointing" means that these varieties have a single point in common, viz., the secret point. Clearly, knowledge of V_d is insufficient to reconstruct the secret and requires activating information, namely

V_i. The scheme proposed in this paper is based on Simmons's preposition secret sharing.

PSS based on Hierarchical Secret Sharing (HSS). In hierarchical secret sharing schemes [24,3,15] the secret is distributed to a group of participants that are partitioned into different hierarchy levels. The reconstruction of the scheme requires to have a certain number of participants from each hierarchy level. An HSS may be configured to address the safeguarding problem by setting a hierarchy featuring the custodians on one level and the user on another. However, a prominent feature required from an HSS in this scenario is prohibit levels from secret reconstruction without communication with another level. Notably, Tassa's scheme allows such configuration by cleverly using polynomial derivatives. To simulate the example in Section 3, let, in the notation of [24], the hierarchical secret sharing be defines with $(\{2, 2\})$ with $\mathcal{U}_0 = \{\text{user\_share}_1, \text{user\_share}_2\}$ and $\mathcal{U}_1 = \{\text{custodian}_1, \text{custodian}_2\}$. The degree of the enabling polynomial in Tassa's HSS is $|\mathcal{U}_0| + |\mathcal{U}_1| - 1 =: 3$, which aligns with our example. In Tassa's HSS, to recover the secret, one must perform the Birkhoff interpolation which admits a unique solution under conditions studied in [24]. Our scheme is conceptually simpler and relies on solving of well-defined systems of linear equations and thus places no such constraints.

6 Discussion

The PSS scheme proposed in this paper is generic by nature and has to be tailored to a specific application. We explain its intrinsic levels of flexibility in Section 2.2 (see, in particular, Fig. 1) and exhaustively describe potential usage modes in Section 4. In this section, we discuss possible implementation concerns.

The setup of the scheme presents a common problem of secure generation and distribution of initializing information to the parties which requires a trusted dealer [12,19]. In a centralized setup, there are two possible candidates for trusted dealer:

- *a trusted third party* creates secret shares and distributes them to the corresponding parties through a secure communication channel. Existence of a party enjoying the trust of all participants is a very heavy assumption. Thus, a verifiable secret sharing scheme as in [12] should be used to assure validity of secret shares.
- *the user*, as the ultimate owner of the secret, bears all the responsibility of parameters' generation (and, possibly, validation) and distribution. This option places heavy requirements on the user's technological capabilities.

Alternatively, similar to [19], in a decentralized setup, each participant creates their shares, distributes them to other participants, and proves the validity of secret shares. Every participant computes their secret shares by combining the shares that they received from other participants. Although this approach incurs more computation and communication costs from each participant, it eliminates the risk of a malicious third party and creates no unnecessary demands for the user.

We have identified the variant with activation on the custodian side as more viable, but left assignment of activators out of scope. As explained in Section 2, a set of activators is provided with points on the instrumental polynomial to reconstruct the secret. To prevent the loss or theft of the secret, redundant activating shares (insufficient by themselves for the activation) must be distributed. To ensure a timely secret reconstruction, such shares must be stored separately from the active shares. A feasible solution is to keep the redundant shares with another custody service, such as a bank. In this option, even if the holder of the redundant shares colludes with the other custodians, they will not be able to reveal the secret since the user's activating share is still required for a full recovery.

7 Conclusion

In this work, being motivated to develop a simple, robust approach to the custodian problem, we proposed a new prepositioned secret sharing scheme that gives a user control over their secret while offering them a distributed custodian service for the protection of the secret. The scheme is conceptually uncomplicated and is based on selective sharing of polynomial coefficients and generation of points on that polynomial which serve as activating information. The proposed PSS scheme provides flexibility regarding the number of parties holding the shares that allows infinite custodian robustness. Furthermore, it prevents the reconstruction of the secret by any subset of parties without the user's activation information.

Our method can be used in custodian applications such as safeguarding a master password to a password vault, a private key for cryptocurrencies' wallets, payment authorization, etc. Additionally, as HSS's can be tailored for use in the custody scenario, our method is germane to simpler HSS scenarios with two level hierarchies with an additional benefit of being extremely concise.

References

1. AMIS: Hierarchical threshold signature scheme (April 2020), `https://github.com/getamis/alice`
2. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: Providing Sound Foundations for Cryptography, pp. 351–371. ACM (2019)
3. Brickell, E.F.: Some ideal secret sharing schemes. In: Workshop on the Theory and Application of of Cryptographic Techniques. pp. 468–475. Springer (1989)
4. Canetti, R., Makriyannis, N., Peled, U.: UC non-interactive, proactive, threshold ECDSA. IACR Cryptol. ePrint Arch. **2020**, 492 (2020)
5. Chaum, D., Crépeau, C., Damgård, I.: Multiparty unconditionally secure protocols (extended abstract). In: STOC. pp. 11–19. ACM (1988)

6. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: TFHE: fast fully homomorphic encryption over the torus. J. Cryptol. **33**(1), 34–91 (2020)
7. Desmedt, Y.: Some recent research aspects of threshold cryptography. In: ISW. Lecture Notes in Computer Science, vol. 1396, pp. 158–173. Springer (1997)
8. Doerner, J., Kondi, Y., Lee, E., Shelat, A.: Threshold ECDSA from ECDSA assumptions: The multiparty case. In: IEEE Symposium on Security and Privacy. pp. 1051–1066. IEEE (2019)
9. Eskicioglu, A.M.: A key transport protocol based on secret sharing - an application to message authentication. In: Communications and Multimedia Security. IFIP Conference Proceedings, vol. 192. Kluwer (2001)
10. Eskicioglu, A.M.: A prepositioned secret sharing scheme for message authentication in broadcast networks. In: Communications and Multimedia Security Issues of the New Century, pp. 363–373. Springer (2001)
11. Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. IACR Cryptol. ePrint Arch. **2012**, 144 (2012)
12. Feldman, P.: A practical scheme for non-interactive verifiable secret sharing. In: FOCS. pp. 427–437. IEEE Computer Society (1987)
13. Gennaro, R., Goldfeder, S.: Fast multiparty threshold ecdsa with fast trustless setup. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. pp. 1179–1194 (2018)
14. Gennaro, R., Goldfeder, S.: One round threshold ECDSA with identifiable abort. IACR Cryptol. ePrint Arch. **2020**, 540 (2020)
15. Ghodosi, H., Pieprzyk, J., Safavi-Naini, R.: Secret sharing in multilevel and compartmented groups. In: Australasian Conference on Information Security and Privacy. pp. 367–378. Springer (1998)
16. Lindell, Y.: Fast secure two-party ECDSA signing. In: CRYPTO (2). Lecture Notes in Computer Science, vol. 10402, pp. 613–644. Springer (2017)
17. Lindell, Y., Pinkas, B.: Secure multiparty computation for privacy-preserving data mining. J. Priv. Confidentiality **1**(1) (2009)
18. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 1592, pp. 223–238. Springer (1999)
19. Pedersen, T.P.: A threshold cryptosystem without a trusted party (extended abstract). In: EUROCRYPT. Lecture Notes in Computer Science, vol. 547, pp. 522–526. Springer (1991)
20. Rabin, T., Ben-Or, M.: Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In: STOC. pp. 73–85. ACM (1989)
21. Shamir, A.: How to share a secret. Communications of the ACM **22**(11), 612–613 (1979)
22. Simmons, G.J.: How to (really) share a secret. In: Conference on the Theory and Application of Cryptography. pp. 390–448. Springer (1988)
23. Simmons, G.J.: Prepositioned shared secret and/or shared control schemes. In: Workshop on the Theory and Application of of Cryptographic Techniques. pp. 436–467. Springer (1989)
24. Tassa, T.: Hierarchical threshold secret sharing. In: TCC. Lecture Notes in Computer Science, vol. 2951, pp. 473–490. Springer (2004)
25. Xu, X., Dexter, S.D., Eskicioglu, A.M.: A hybrid scheme for encryption and watermarking. In: Security, steganography, and watermarking of multimedia contents VI. vol. 5306, pp. 725–736. International Society for Optics and Photonics (2004)